

EXPRESS MAIL MAILING LABEL NO. EU684674717US

PATENT APPLICATION  
ATTORNEY DOCKET NO. 20040189-US-NP

## COLOR DEVICE PROFILE HAVING A BUFFERED LOOK-UP TABLE

### BACKGROUND

- [0001]** The subject disclosure is generally directed to printer profiles.
- [0002]** Color device profiles are commonly employed to transform color image data formatted for one device color space to color image data formatted for another device color space. For example, an International Color Consortium (ICC) workflow calls for converting color image data formatted in red-green-blue (RGB) color space for a color monitor to a device independent color space called profile connection space (PCS), and then into color image data formatted in CMYK space for a color printer.
- [0003]** As a result of computational limitations, it may be difficult to produce clean primary and/or secondary output colors.

### BRIEF DESCRIPTION OF DRAWINGS

- [0004]** FIG. 1 is a schematic block diagram of an embodiment of a computer system in which the disclosed color transform pipeline can be employed.
- [0005]** FIG. 2 is a schematic block diagram of an embodiment of a color transform pipeline.

**[0006]** FIG. 3 is a schematic illustration of an embodiment of a 2-dimensional buffered look-up table wherein the outermost nodes are replicas of inner nodes that are closest to the outermost nodes.

**[0007]** FIG. 4 is a schematic illustration of an embodiment of a 3-dimensional buffered look-up table wherein the outermost nodes are replicas of inner nodes that are closest to the outermost nodes.

**[0008]** FIG. 5 is a schematic diagram illustrating a buffered input range for a buffered color look-up table.

**[0009]** FIG. 6 is a schematic flow diagram of an embodiment of a procedure for generating a buffered color look-up table and scaled input tonal reproduction curves.

## DETAILED DESCRIPTION

**[0010]** FIG. 1 is a schematic block diagram of an embodiment of a computer system 10 in which the disclosed color transform pipeline can be utilized. The computer system includes a processor 11, a read-only memory 13, random access memory 15, a mass storage device 17 such as a disk drive, and a removable storage device 19 such as a DVD drive. The computer system further includes a display 21 which can be a color monitor, for example, an input device 23 such as a keyboard or touch pad for entering user commands, a pointing device 25 such as a mouse or trackball for pointing to and manipulating graphical user interfaces or other objects displayed on the display 21. The computer system 10 can be connected to a printer 27, for example via a local area network. The computer system 10 can also be connected to a scanner 29.

**[0011]** FIG. 2 is a schematic block diagram of an embodiment of a color transform pipeline that can be employed to map or transform source device dependent color space SCS input values to output destination device dependent color space DSC output values. The source device color space SCS can be formatted or configured for a display device such as a color monitor, while the

destination device color space DCS can be formatted or configured for a hardcopy apparatus such as a printer. An example of source device color space would be RGB (red-green-blue), while an example of a destination device color space would be CMYK (cyan-magenta-yellow-black). Thus, for example, RGB input color data formatted for a color monitor is mapped or transformed to CMYK output color data that is formatted for a printer.

**[0012]** The color transform pipeline can be implemented in a color management module (CMM) of computer operating system or an application, for example in accordance with an ICC workflow. For ease of reference, this disclosure will be in the context of source device dependent RGB color space and destination device dependent CMYK color space, and it should be appreciated that other color spaces can be employed.

**[0013]** The color transform pipeline includes a device to PCS profile 30 that maps or converts source device dependent color space input values such as RGB values to profile connection space (PCS) values XYZ. The device to PCS profile 30 can employ for example tonal reproduction curves (TRCs) 31 and a 3X3 matrix 33. The TRCs 31 provide adjusted source device color space values that are multiplied by the matrix 33 to produce the PCS XYZ values. The device to PCS profile 30 can be a device specific profile (e.g., for a particular scanner or monitor), or it can be a quasi-standard device color space such as sRGB.

**[0014]** The color transform pipeline further includes a PCS to device profile 40 that maps or converts the PCS values to destination device dependent color output values such as CMYK. The PCS to device profile 40 includes a 3X3 matrix 41, input TRCs 43, an interpolator 45, and output TRCs 47. The 3x3 matrix 41 maps the PCS values to an intermediate destination device color space ICS, the input TRCs 43 adjust the intermediate destination device color values and provides inputs to the interpolator 45 which maps the adjusted intermediate color values to destination device color space values such as CMYK. The destination device color space values

can be adjusted pursuant to output TRCs 47 to provide adjusted destination device color space values.

**[0015]** The interpolator 45 employs a color look-up table 145 that contains destination device color values such as CMYK. Each CMYK value in the table can be called a node and comprises the CMYK value for a corresponding location in the TRC adjusted intermediate color space ICS provided by the input TRCs 43. Values between nodes (i.e., CMYK values for adjusted PCS values between the PCS values associated with the nodes) are calculated using interpolation. The PCS to device profile 40 can be specific to a particular printer, for example, and generates the appropriate destination device color space values (e.g., CMYK) that will cause the output device to produce the colors specified by the PCS XYZ data. Since the outputs of the input TRCs are utilized by the interpolator 45 to obtain data from the color look-up table 145, it can be convenient to refer to the outputs of the TRCs as inputs to the color look-up table, and the range of the values of the TRCs can be called an input range. The input TRCs can be implemented as tables and can be called input tables.

**[0016]** The color space of the color look-up table 145 of the PCS to device profile 40 can be aligned with the source device color space SCS, for example wherein the input range of the interpolator 45 corresponds to the range of the source device color space SCS inputs. This can be accomplished, for example, by configuring the 3X3 matrix 41 of the destination device profile 40 to be the inverse of the 3X3 matrix 33 of the source device profile 30, and configuring the input TRCs 43 to be the inverse of the source device TRC's 31. In this manner, the PCS values are multiplied by a matrix 41 that is the inverse of the matrix 33, and results of the multiplication by the matrix 41 are adjusted or linearized by input TRCs 43 that are the inverse of the source device TRC's 31.

**[0017]** The outputs of the input TRCs 43 thus approximately correspond to the input RGB values, comprising for example approximate replicas of the input RGB values or approximate scaled replicas of the input RGB values. In other words, the

outputs of the input TRCs 43 can comprise the input RGB values with some error terms introduced by the processing or scaled versions of the input RGB values with some error terms introduced by the processing. By way of illustrative example, the input RGB values can be 8-bit data, while the values of the input TRCs can be 8-bit or 16-bit data. The color look-up table 145 can thus comprise a look-up table that maps RGB values to CMYK values, for example.

**[0018]** The color look-up table 145 of the destination device profile 40 can more particularly be configured as an expanded or buffered color look-up table wherein data contained in the outermost grid points or nodes are replicas of the next to outermost nodes (i.e., the inner nodes that are closest to the outermost nodes). A buffered color look-up table 145 can comprise an inner region or portion and an outer region or portion, wherein the outer portion surrounds the inner portion and comprises replicas of the closest nodes of the inner portion. In other words, the nodes of the outer portion are replicas of the outermost nodes of the inner portion. By way of illustrative example, the outer portion comprises one layer of nodes, such that the indexes or addresses associated with the nodes of the outer portion are spaced from the inner portion outermost nodes they replicate by one grid interval INT along one or more axes, depending on the location of a particular inner portion outermost node.

**[0019]** FIG. 3 is a schematic depiction of a 2-dimensional buffered look-up table 245 that includes an inner portion 245A comprising nodes on or inside a generally parallelogram shaped reference boundary 245B and an outer portion 245C comprising nodes outside the generally parallelogram shaped boundary 245B and surrounding the inner portion 245A. The nodes of the outer portion 245C replicate the outermost nodes of the inner portion 245A as indicated by arrows leading from outermost nodes of the inner portion 245A to replicating nodes of the outer portion 245C. In other words the data contained in the outermost nodes of the inner portion 245A are replicated in the nodes of the outer portion 245C.

**[0020]** FIG. 4 is a schematic perspective representation of a 3-dimensional buffered color look-up table 345 as a 3-dimensional rectilinear grid that includes an inner portion 345A comprising nodes on or inside an inner rectilinear reference box 345B and an outer portion 345C comprising nodes on an outer rectilinear reference box 345D that surrounds the inner portion 345A. The nodes of the inner portion 345A are illustrated as hollow dots, while the nodes of the outer portion are illustrated as solid dots. The solid and hollow dots become smaller with distance from the front surface of the outer rectilinear box 345D. The inner portion nodes that are closest to the nodes of the outer portion 345C are on the rectilinear box 345B and are replicated in the nodes of the outer portion 345C. More particularly, each node at a corner of the inner rectilinear box 345B is replicated at 7 nodes in the outer portion, each node along an edge between corners of the inner rectilinear box 345B is replicated at 3 nodes in the outer portion, and each node on a surface of the inner rectilinear box 345B between edges and corners is replicated at 1 node in the outer portion. A buffered 3-dimensional  $G \times G \times G$  color look-up table thus would have an  $(G-2) \times (G-2) \times (G-2)$  inner portion, for the illustrative example of a single layer outer or buffer portion. The outer portion would have  $[G \times G \times G - (G-2) \times (G-2) \times (G-2)]$  nodes.

**[0021]** The indexes or addresses associated with the nodes along each axis of a buffered color look-up table 145 can be 0 to MAX, for example with a substantially constant or uniform grid interval of INT. MAX can be a positive integer and can typically be  $2^N - 1$ , wherein N can be 8 or 16 or 24, for example, depending on the color transform pipeline. For the illustrative example wherein the outermost nodes comprise replicas of the closest inner nodes, a replicated node would have an associated index of  $0 + \text{INT}$  or  $\text{MAX} - \text{INT}$  along at least one axis, while a replicating node would have an associated index of 0 or MAX along at least one axis, as schematically depicted in FIG. 5 for a representative line of nodes 445 along one dimension of the look-up table.

**[0022]** The input TRCs 43 of the PCS to device profile 40 can be configured to provide inputs to the interpolator 45 that are configured such that the TRC values

have a range along each dimension that is less than the look-up table index range of 0 to MAX and greater than the index range of  $0+INT$  to  $MAX-INT$  for the inner portion of the look-up table. For example, the TRC values can have a range in each dimension of  $(0+INT-E0)$  to  $(MAX-INT+E1)$ , wherein each of  $E0$  and  $E1$  can be a relatively small number that is less than the look-up table grid interval  $INT$ , for example. This concept is schematically illustrated in FIG. 5 for a representative line of nodes 445 along one dimension of the color look-up table. The line comprises inner portion nodes 445A disposed between outer portion nodes 445C. The inner portion nodes 445A comprises a line of nodes of the inner portion of the buffered color look-up table along one dimension, while the outer portion nodes 445C comprise nodes of the outer portion of the buffered color look-up table along one dimension. The outer portion nodes 445C replicate the inner portion nodes 445A that are closest to the outer portion nodes 445C. Such closest nodes are the inner portion nodes 445C that are immediately adjacent the outer portion nodes 445C. The index range for the entire line of nodes is from 0 to MAX, while the index range of the inner portion nodes is from  $0+INT$  to  $MAX-INT$ , wherein  $INT$  is the grid interval.

**[0023]** Alternatively, the TRC values can have a range in each dimension of 0 through  $(MAX-INT+E1)$ , or  $(0+INT-E0)$  through MAX. In this manner, the input range (i.e., the range of the TRC values) overlaps the inner portion of the look-up table.

**[0024]** Since the ranges of the input TRCs are smaller than the index ranges of the buffered color look-up table 145 and overlap the inner portion of the buffered color look-up table 145, calculation errors are buffered such that requests for a primary output color tends to consistently result in a primary output color. This is because an input to the interpolator that is outside the inner portion of the buffered color look-up table will produce an interpolator output that is the same output for an input that is on the outer surface of the inner portion of the buffered color look-up table.

**[0025]** The re-scaled input TRCs 43 and the buffered color look-up table 145 can be generated by first generating input TRCs and a color look-up table that are

not buffered and have an index range of 0 to MAX. The color look-up table is expanded to include outermost nodes that replicate what were outermost nodes in the originating color look-up table, while the input TRCs are scaled to have a range that extends slightly beyond the nodes of the expanded or buffered look-up table that correspond to the original color look-up table.

**[0026]** FIG. 6 sets forth a schematic flow diagram of an embodiment of a procedure for generating a buffered GxGxG color look-up table (CLUT) from a  $(G-2) \times (G-2) \times (G-2)$  color look-up table that is configured for example to map input RGB values to output CMYK values, and for scaling input TRCs. The originating CLUT is stored in an array or table L, while the buffered CLUT is stored in an array or table L'. The originating input TRCs are in input arrays or tables Tu' while the scaled TRCs are in input arrays or tables Tu. Each of the originating and scaled input TRCs includes t values.

**[0027]** At 112 the number of nodes of the originating CLUT along each axis or dimension is specified by the variable G'.

**[0028]** At 113 n is set to the number of input axes for the originating and buffered CLUTs which is 3 for this embodiment, m is set to the number of output axes (i.e., the output dimension), and the originating CLUT is stored as a single dimensional array L' of size  $(G')^3 m$ . Each element in the array L' is a non-negative scalar value.  $T_i$  is the originating input TRC of t values for the i'th axis, and each originating TRC value can be a non-negative scalar value less than or equal to the constant MAX.

**[0029]** At 114, G is set to the number of nodes per axis for the buffered CLUT, and the first index counter i for the buffered CLUT is initialized to 0. The buffered CLUT will have an extra 2 nodes per axis and thus  $G = G' + 2$ .

**[0030]** At 115 the value of i' is computed from i using the equation  $i' = \text{Max}(0, \text{Min}(G' - 1, i - 1))$ , wherein i' is the first index counter for the originating CLUT. Also, the second index counter j for the buffered CLUT is initialized to 0.



**[0031]** At 116 the value of  $j'$  is computed from  $j$  using the equation  $j' = \text{Max}(0, \text{Min}(G'-1, j-1))$ , wherein  $j'$  is the second index counter for the originating CLUT. Also, the third index counter  $k$  for the buffered CLUT is initialized to 0.

**[0032]** At 117 the value of  $j'$  is computed from  $j$  using the equation  $j' = \text{Max}(0, \text{Min}(G'-1, j-1))$ , wherein  $j'$  is the second index counter for the originating CLUT. Also, the output dimension counter  $u$  is initialized to 0. The originating and buffered CLUTs have the same output dimension, and therefore will use the same output dimension counter.

**[0033]** At 118 the buffered CLUT  $L$  is populated. The inner nodes of the originating CLUT are copied into the buffered CLUT only once, while nodes on the outer surface of the originating CLUT are copied into the buffered CLUT multiple times. The buffered CLUT array  $L$  can be a single dimensional array of size  $(G)^3m$ . The arrays  $L$  and  $L'$  can be accessed through respective multi-dimensional hash functions  $a$  and  $a'$ , for example in accordance with an industry standard packing order.

**[0034]** At 119 the output dimension counter  $u$  is incremented by 1.

**[0035]** At 120  $u$  is compared to  $m$ . This is the flow control for the output dimension counter loop. If  $u < m$ , processing continues at 118. Otherwise, processing continues at 121.

**[0036]** At 121 the third index counter  $k$  is incremented by 1.

**[0037]** At 122  $k$  is compared to  $G$ . This is the flow control for the third index counter loop. If  $k < G$ , processing continues at 117. Otherwise, processing continues at 124.

**[0038]** At 124 the second index counter  $j$  is incremented by 1.

**[0039]** At 123,  $j$  is compared to  $G$ . This is the flow control for the second index counter loop. If  $j < G$ , processing continues at 116. Otherwise, processing continues at 125.

**[0040]** At 125 the first index counter  $i$  is incremented by 1.

**[0041]** At 126,  $i$  is compared to  $G$ . This is the flow control for the first index counter loop. If  $i < G$ , processing continues at 115. Otherwise, processing continues at 127.

**[0042]** At 127, the input dimension counter  $u$  is initialized to 0.

**[0043]** At 128, the index counter  $i$  is initialized to 0.

**[0044]** At 130 the input range of the TRCs is adjusted.  $T_0$ ,  $T_1$ , and  $T_2$  are the new input TRCs, each of which can be a one dimensional array of length  $t$ . A scaled input TRC value is computed from an originating input TRC value and placed in the  $T_u$  array.

**[0045]** At 131 the index counter  $i$  is incremented by 1.

**[0046]** At 132,  $i$  is compared to  $t$ . This is the flow control for the index counter loop. If  $i < t$ , processing continues at 130. Otherwise, processing continues at 133.

**[0047]** At 133 the input dimension counter  $u$  is incremented by 1.

**[0048]** At 129  $u$  is compared to  $n$ . This is the flow control for the input dimension counter loop. If  $u < n$ , processing continues at 128. Otherwise, processing is finished.

**[0049]** The claims, as originally presented and as they may be amended, encompass variations, alternatives, modifications, improvements, equivalents, and substantial equivalents of the embodiments and teachings disclosed herein, including those that are presently unforeseen or unappreciated, and that, for example, may arise from applicants/patentees and others.